

21世纪高等学校规划教材 | 计算机科学与技术



面向对象程序设计

况立群 熊风光 杨晓文 等 编著

清华大学出版社

21世纪高等学校规划教材 | 计

面向对象程序设计

况立群 熊风光 杨晓文 等 编著



清华大学出版社
北京

内 容 简 介

面向对象程序设计是计算机科学与技术学科的一门必修课程。面向对象软件开发集抽象性、封装性、继承性和多态性于一体,可以帮助人们开发出模块化、数据抽象程度高、体现信息隐蔽、可复用、易修改、易扩充等特性的程序。

本书以语言简明、概念清晰、逻辑严谨、容易入门为特色,以通俗易懂的语言讲授面向对象程序设计的基础知识、基本规则和编程思想,以贴近生活的例题去表现面向对象程序设计的思想,将面向对象程序设计思想与 C++ 语言有机地结合在一起。

本书内容丰富,是跨编程基础、过程化编程、对象化编程的三合一。教师可以参考学生的已修课程,或选择其中的第一、二部分讲授面向过程程序设计,或选择第三部分讲授面向对象程序设计。

本书既可为没有 C++ 基础的人员学习基础 C++ 提供入门之需,也可为具有一定 C++ 语言的人员深入学习面向对象程序设计的知识提供帮助,可作为大专院校计算机专业和非计算机专业学生学习面向对象程序设计的教材,也可作为 C++ 爱好者或自学人员的学习 C++ 的参考书。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

面向对象程序设计/况立群,熊风光,杨晓文等编著.--北京:清华大学出版社,2013

21世纪高等学校规划教材·计算机科学与技术

ISBN 978-7-302-32114-9

I. ①面… II. ①况… ②熊… ③杨… III. ①面向对象语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 082895 号

责任编辑:郑寅堃 赵晓宁

封面设计:傅瑞学

责任校对:白 蕾

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:18.75 字 数:454千字

版 次:2013年6月第1版 印 次:2013年6月第1次印刷

印 数:1~2000

定 价:33.00元

产品编号:045228-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

- (1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。
- (2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。
- (3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。
- (4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。
- (5) 21 世纪高等学校规划教材·信息管理与信息系统。
- (6) 21 世纪高等学校规划教材·财经管理与应用。
- (7) 21 世纪高等学校规划教材·电子商务。
- (8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人: 魏江江

E-mail: weijj@tup.tsinghua.edu.cn

前言

目前,我国计算机的发展仍落后于发达国家,中国要成为软件产业大国,必须培养大量的高级程序员和软件工程师,从事面向对象的程序设计开发工作。国内外高校计算机专业均设置了面向对象程序设计课程,并且采用了经典的 C++ 语言进行教授,对 C++ 语言的学习正如火如荼地展开。学习 C++ 语言,不光是要学习和掌握如何使用语言本身,更重要的是掌握和理解面向对象的编程思想,能够进行面向对象的分析(OOA)和面向对象的设计(OOD),最后用 C++ 进行面向对象的编程(OOP),自然也就水到渠成了。鉴于此,我们决定编著一本基于 C++ 的面向对象程序设计教材。

本书分三大部分,共 11 章。其中第一部分为编程基础,包括第 1 章面向对象与 C++ 语言概述,第 2 章数据类型,第 3 章运算符及表达式,第 4 章语句及预处理命令;第二部分为面向过程编程,包括第 5 章数组、指针、引用和字符串,第 6 章函数;第三部分为面向对象编程,包括第 7 章类与对象基础,第 8 章类与对象深入,第 9 章运算符重载,第 10 章继承与派生,第 11 章多态性与虚函数。

与国内外同类书相比,本书具有如下特色和价值:

(1) 全书以递进方式展开。

全书重点介绍了编程基础、面向过程编程、面向对象编程三个方面的内容,内容难度上层层递进,通过这种安排,系统的介绍面向对象程序设计的思想与面向过程的思想的不同,让学生掌握面向对象软件开发的流程。

(2) 授课时数有可参考性。

本书内容是跨编程基础、过程化编程、对象化编程的三合一。选用该教材的教师可以参考学生的已修课程,选择进行其中的第一、二部分或第三部分的教学。

(3) 代码特色。

书中的代码从性能和易读风格上力求精简,并且书中的所有代码来源于编者所参与的实际项目,以实际案例为导向,更能吸引学生的兴趣。

本书第 1 和第 2 章由韩燮编写,第 3、第 9 和第 11 章由熊风光编写,第 4 和第 10 章由杨晓文编写,第 5 和第 8 章由况立群编写,第 6 和第 7 章由张元编写。

编者

2013 年 5 月

目 录

第一部分 编程基础

第 1 章 面向对象与 C++ 语言概述	3
1.1 程序设计语言	3
1.1.1 机器语言与汇编语言.....	3
1.1.2 高级语言.....	4
1.1.3 面向过程的语言.....	4
1.1.4 面向对象的语言.....	4
1.2 C++ 的起源及特点	6
1.2.1 C++ 的起源	6
1.2.2 C++ 的特点	6
1.3 C++ 程序的开发过程	7
1.4 C++ 的词法与语法规则	8
1.4.1 字符集.....	8
1.4.2 标识符.....	8
1.4.3 关键字.....	9
1.4.4 分隔符.....	9
1.4.5 注释符.....	9
1.5 C++ 程序结构	10
1.5.1 简单的 C++ 程序	10
1.5.2 C++ 程序的结构	12
1.6 Visual C++ 简介	13
1.6.1 编辑 C++ 源程序	13
1.6.2 编译和连接	18
1.6.3 运行	20
习题 1	20
第 2 章 数据类型	22
2.1 基本数据类型.....	22
2.2 字面值常量.....	23
2.2.1 字面值整型常量	23
2.2.2 字面值字符常量	24

2.2.3	字面值浮点常量	24
2.2.4	字面值布尔常量	25
2.2.5	字面值字符串常量	25
2.3	变量	25
2.3.1	变量名	25
2.3.2	变量的定义	26
2.3.3	变量的初始化	26
2.4	符号常量	27
2.5	枚举类型	27
	习题 2	29
第 3 章 运算符及表达式		31
3.1	什么是表达式	31
3.2	算术运算符与算术表达式	31
3.3	自增和自减运算符与自增和自减表达式	32
3.4	赋值运算符与赋值表达式	32
3.5	关系运算符与关系表达式	33
3.6	逻辑运算符与逻辑表达式	33
3.7	位运算	34
3.8	条件运算符与条件表达式	35
3.9	sizeof 运算符	36
3.10	逗号运算符与逗号表达式	36
3.11	运算符的优先级和结合性	36
3.12	数据类型转换	38
3.12.1	隐式转换	38
3.12.2	显式转换	38
	习题 3	39
第 4 章 语句及预处理命令		42
4.1	简单语句和复合语句	42
4.1.1	简单语句	42
4.1.2	复合语句	43
4.2	选择语句	43
4.2.1	if 语句	44
4.2.2	switch 语句	48
4.3	循环语句	52
4.3.1	for 语句	52
4.3.2	while 语句	54
4.3.3	do-while 语句	56

4.3.4	循环的嵌套	57
4.4	转向语句	59
4.4.1	break 语句	59
4.4.2	continue 语句	60
4.4.3	goto 语句	61
4.5	预处理命令	61
4.5.1	宏定义	61
4.5.2	文件包含命令	65
4.5.3	条件编译命令	65
习题 4	67
第二部分 面向过程编程		
第 5 章 数组、指针、引用和字符串		
5.1	数组	75
5.1.1	一维数组	75
5.1.2	二维数组	78
5.1.3	字符数组	81
5.2	指针	82
5.2.1	指针的定义	82
5.2.2	指针的赋值和初始化	83
5.2.3	取指针指向的变量	83
5.2.4	指针运算	84
5.3	指针与数组	85
5.3.1	指针与一维数组	85
5.3.2	指针与二维数组	87
5.4	引用	88
5.5	字符串	92
5.5.1	C 风格字符串	92
5.5.2	string 类型字符串	94
习题 5	98
第 6 章 函数		
6.1	函数概述	103
6.2	函数的定义、调用和声明	103
6.2.1	函数的定义	103
6.2.2	函数的调用	104
6.2.3	函数的声明	105
6.3	函数的参数传递和返回值类型	106

6.3.1	值传递	106
6.3.2	引用传递	107
6.3.3	函数的返回值类型	108
6.4	函数重载	110
6.5	函数参数的缺省值	112
6.6	内联函数	113
6.7	作用域	114
6.7.1	局部域	114
6.7.2	全局变量	116
6.7.3	局部变量	116
6.8	异常处理	118
	习题 6	120

第三部分 面向对象编程

第 7 章	类与对象基础	129
7.1	面向对象程序设计概述	129
7.1.1	类与对象	129
7.1.2	可扩展性与可重用性	130
7.2	类的定义与声明	132
7.2.1	类的定义语法	132
7.2.2	类的访问说明符	132
7.2.3	数据成员	133
7.3	成员函数的定义	133
7.3.1	成员函数	133
7.3.2	设置和获取数据	135
7.4	对象的定义	135
7.5	构造函数	138
7.5.1	特殊函数	138
7.5.2	构造函数	138
7.6	析构函数	139
7.7	拷贝构造函数	139
7.8	格式	143
7.8.1	隐藏函数定义	143
7.8.2	#include "header. h"	146
7.8.3	条件编译	146
	习题 7	147

第 8 章 类与对象深入	153
8.1 常对象和 const 成员函数	153
8.2 成员初始化	158
8.2.1 成员初始化列表	158
8.2.2 const 数据成员与成员初始化列表	159
8.3 静态成员	160
8.3.1 静态成员函数	162
8.3.2 静态数据成员	162
8.3.3 常静态数据成员	163
8.4 友元	163
8.5 对象数组	167
8.6 对象指针	169
8.7 this 指针	171
8.8 指向成员的指针	172
8.8.1 指向数据成员的指针	172
8.8.2 指向成员函数的指针	174
8.9 堆对象	176
8.9.1 动态内存分配和销毁	177
8.9.2 内存块的动态分配和销毁	178
8.9.3 对象/对象数组的动态创建和销毁	179
习题 8	182
第 9 章 运算符重载	187
9.1 简介	187
9.2 运算符重载基础	187
9.3 运算符重载的限制	188
9.4 运算符重载为成员函数	189
9.5 运算符重载为友元函数	194
9.6 赋值运算符的重载与深拷贝、浅拷贝	199
习题 9	209
第 10 章 继承与派生	211
10.1 简介	211
10.2 继承引例	211
10.3 公有继承	214
10.3.1 派生类对象的构造过程	220
10.3.2 派生类对象的析构过程	221
10.4 保护的访问说明符	221

10.5	私有继承和单继承与多继承	225
10.5.1	私有继承	225
10.5.2	单继承与多继承	229
10.6	保护继承	230
10.7	组合	231
10.8	虚拟继承——二义性的解决方案	235
10.9	继承或组合	240
10.10	组合和聚合	241
习题 10	241
第 11 章	多态性与虚函数	249
11.1	程序设计语言中的多态	249
11.2	子类型与赋值兼容原则	250
11.2.1	子类型	250
11.2.2	赋值兼容	250
11.3	虚函数	256
11.3.1	虚函数的定义	258
11.3.2	重写	258
11.3.3	虚函数实现动态多态的机制	260
11.3.4	构造函数、析构函数、成员函数中调用虚函数	262
11.4	虚析构函数	264
11.5	纯虚函数和抽象类	267
11.5.1	纯虚函数	268
11.5.2	抽象类	269
11.6	综合案例——工资管理系统	273
习题 11	276
参考文献	285

第①部分

编程基础

- 第1章 面向对象与C++ 语言概述
- 第2章 数据类型
- 第3章 运算符及表达式
- 第4章 语句及预处理命令

第 1 章

面向对象与C++语言概述

C++语言是一种应用广泛的程序设计语言,是在C语言基础上发展起来的并且加入了面向对象的支持,从而成为一种面向对象的程序设计语言。

本章首先介绍程序设计语言的发展经历,重点是面向对象的程序设计语言及其相关概念;然后介绍面向对象语言C++的起源和特点、程序开发过程、词法和语法规则、程序结构;最后重点介绍如何利用如何Microsoft的集成开发工具Visual C++6.0开发C++程序。

1.1 程序设计语言

语言是人类独有的以言语形式进行表达和交流的一种具有词法和语法规则的系统,是客观世界与人的主观条件相结合的产物。

计算机程序设计语言,通常简称为编程语言,具有一组用来定义计算机程序的语法规则。它是一种被标准化的交流技巧,用来向计算机发出指令,让程序员能够准确地定义计算机所需要使用的数据,并精确地定义在不同情况下应当采取的行动。

1.1.1 机器语言与汇编语言

机器语言是直接由二进制代码指令表达的计算机语言,指令是用0和1组成的一串代码,它们有一定的位数,并分成若干段,各段的编码表示不同的含义。例如,某台计算机字长为16位,即有16个二进制数组成一条指令或其他信息。16个0和1可组成各种排列组合,通过线路变成电信号,让计算机执行各种不同的操作。因此,机器语言是特定计算机系统所固有的语言。用机器语言进行程序设计,需要对机器结构有较多的理解。而且,用机器语言进行软件开发的难度大、周期长,编制出来的程序可读性很差,难以维护和修改。

为了提高程序设计的效率,人们考虑用有助记忆的符号来表达机器指令中的操作码和操作数,如用ADD表示加法,SUB表示减法等,这就是汇编语言(Assembly Language),因此汇编语言也称为符号语言。使用汇编语言编写的程序,机器不能直接识别,要由一种程序将汇编语言翻译成机器语言,这种起翻译作用的程序叫汇编程序,汇编程序是系统软件中的语言处理系统软件。汇编程序把汇编语言翻译成机器语言的过程称为汇编。

汇编语言虽然比机器语言易于读写、调试和修改,但是汇编语言仍然还是一种层次非常

低的语言,只仅仅高于直接手工编写二进制的机器指令码而已,对于人们抽象思维和交流仍然十分不便。

1.1.2 高级语言

高级语言是较接近自然语言和数学公式的程序语言,与计算机的硬件结构及指令系统无关,具有更强的表达能力,可方便地表示数据的运算和程序的控制结构,能更好地描述各种算法,让人们能以更易理解的方式编写程序,提升了语言的抽象层次。

高级语言并不是特指某一个具体的语言,而是包括很多编程语言,如目前流行的 C、Pascal、Basic、C++ 等,这些语言的语法、命令格式都不尽相同。

1.1.3 面向过程的语言

面向过程的语言也称为结构化程序设计语言,是高级语言的一种。在面向过程程序设计中,问题被看作一系列需要完成的任务,函数则用于完成这些任务,解决问题的焦点集中于函数。其概念最早由 E. W. Dijkstra 在 1965 年提出,是软件发展的一个重要里程碑。它的主要观点是采用自顶向下、逐步求精的程序设计方法,使用三种基本控制结构构造程序,即任何程序都可用顺序、选择、循环三种基本控制结构构造。

面向过程的语言也存在一些问题:

- (1) 用户需求难以在系统分析阶段准确定义,致使系统在交付使用时产生许多问题。
- (2) 对系统开发每个阶段的成果进行严格控制,不能适应事物变化的要求。
- (3) 系统的开发周期长。

使用面向过程的语言进行编程的程序设计方法,称为面向过程的程序设计。

1.1.4 面向对象的语言

面向对象的语言是比面向过程的语言更高级的一种高级语言。面向对象的语言是以人类认识客观世界为出发点,描述客观世界中存在的事物以及它们之间的关系。它的出现改变了程序员的思维方式,使得程序设计的出发点由着眼于问题域中的函数转向问题域中的对象及其相互关系。因此,面向对象的语言更接近于人类的自然语言,体现了人们对于客观事物的更高层次的抽象。

1. 基本概念

面向对象的语言中,一般包含如下概念。

- (1) 面向对象(object-oriented, OO)是一种运用对象、类、封装、包含、继承、消息传递、多态性等概念来构造系统的软件开发方法。
- (2) 对象(object): 现实世界中客观事物的抽象表示,如张三、苹果、飞机、规则、法律、法规等。每个对象皆有自己的内部特征和行为,如张三具有名字、身高、体重等内部特征,具有吃饭、睡觉、打人、偷税、漏税等行为。在面向对象概念中我们把对象的内部特征称为属性,行为称为方法。那么,也可以称对象为一个属性和方法结合体,对外屏蔽内部的部分特征。对象的属性描述了对象某一时刻的内部特征,而通过对象的方法可以使得对象内部的

特征发生改变,如张三吃饭了以后他的体重这个内部特征就会发生改变,对于张三而言,吃饭前和吃饭后的体重肯定会有所不同。

(3) 类(class): 类是具有相似内部状态和运动规律的对象抽象。类的概念来自于人们认识自然、认识社会的过程,在这一过程中,人们主要使用两种方法:由特殊到一般的归纳法和由一般到特殊的演绎法。在归纳的过程中,从一个个具体的事物中把共同的特征抽取出来,形成一个一般的概念,这就是归类;如昆虫、狮子、爬行动物,因为它们都能动所以归类为动物。在演绎的过程中又把同类的事物,根据不同的特征分成不同的小类,这就是分类;如动物→猫科动物→猫→大花猫等。对于一个具体的类,它有许多具体的个体,将这些个体叫做对象。类的内部状态是指类集中对象的共同状态;类的运动规律是指类集中对象的共同运动规律。如柏拉图对人作如下定义:人是没有毛能直立行走的动物。在柏拉图的定义中人是一个类,具有没有毛、直立行走等一些区别于其他事物的共同特征;而张三、李四、王五、赵六等一个个具体的人,均是这个类的具体对象。

(4) 消息(message): 消息是指对象间相互联系和相互作用的方式。一个消息主要由5部分组成:发送消息的对象、接收消息的对象、消息传递办法、消息内容(参数)、反馈。

2. 基本特征

面向对象语言具有封装性、继承性和多态性三大基本特征。

1) 封装

封装是面向对象编程的特征之一,也是类和对象的主要特征,其中类是封装的工具,而对象则是一个具体的封装实现。封装将数据以及加在这些数据上的操作组织在一起,成为有独立意义的构件。外部无法直接访问这些封装了的数据,从而保证了这些数据的正确性和独立性,实现了对数据的访问控制。反过来说,如果外部可以任意使用一个类的数据,也就可对这个类做任何事情,那么就没有办法强制它们遵守任何约束。

外部访问类里面的数据必须通过接口(interface)进行访问。接口规定了可对一个特定的对象发出哪些请求,进而执行相关的代码,以便满足这些请求。这些代码与那些隐藏起来的数据叫作“隐藏的实现”。站在面向过程程序设计的角度,整个问题并不显得复杂,每种请求都会有一个与之相关联起来的函数,一旦向对象发出一个特定的请求,就会调用一个与请求相关联的函数。通常将这个过程总结为向对象发送一条消息。对象的职责就是决定如何对这条消息作出反应,也就是找到相应的代码并执行。

有两方面的原因促使了类对其数据进行访问控制。第一个原因是防止用户(类的使用者)接触他们不该接触的数据。若只是为了解决特定问题,需要使用这些不该接触的数据,用户只需操作接口即可,避免对这些数据的直接访问(这些访问可能会导致一些流程控制的错误)。类向用户提供的实际上是一种服务,用户只需要知道类是否满足了其功能需求,而不必关心这些功能是如何实现的。第二个原因是允许类的创建者修改内部结构,而不会对用户编写的程序造成任何影响。例如,创建者最开始可能设计了一个形式简单的类,定义了数据以及接口的简单实现,以便简化开发,日后如若需要增强该类的功能,使其更快地运行,只需要对类中接口的实现进行重写即可,用户程序不会受任何影响。